

In this code, a Windows Forms application is designed to function as a quiz or trivia game. It utilizes various GUI controls such as labels, textboxes, picture box, and radio buttons to create an interactive user experience. The application allows users to answer a series of questions, provides feedback on the correctness of the answers, and displays the final score at the end. Here's a breakdown of the code's functionality:

1. Initialization and Welcome Message:

- The application starts with a welcome message displayed using `MessageBox.Show("Welcome to TriviaMaster! 🎮")` when the form is loaded.

2. Question and Answer Data:

- Questions and corresponding answers are stored in arrays (`questions`` and `answers``).
- The current question index (`currentQuestionIndex``) and the user's score (`score``) are tracked.

3. DisplayQuestion Function:

- The `DisplayQuestion`` function is responsible for setting up the GUI to display the current question, answer options, and difficulty levels.
- It dynamically changes the displayed question, image, and options based on the current question index.

4. Textbox KeyPress Event:

- The `TxtAnswer_KeyPress`` event is triggered when the user presses the Enter key in the answer textbox.
- It checks the entered answer against the correct answer, updates the score, and proceeds to the next question.
- If all questions are answered, a message box shows the final score and prompts the user to try again.

5. Checkbox CheckedChanged Event:

- The ``ChkOption_CheckedChanged`` event is triggered when the checkbox (``chkOption``) is checked.
- It displays the current score in a message box.

6. Radio Button CheckedChanged Events:

- There are three radio button events (``RadEasy_CheckedChanged``, ``RadMedium_CheckedChanged``, ``RadHard_CheckedChanged``) triggered when the user selects a difficulty level.
- Each event displays a message box indicating the selected difficulty. Additional logic for handling difficulty can be added here.

7. Quiz Completion Handling:

- After completing all questions, a message box prompts the user with the final score and an option to try the quiz again.
- If the user chooses to try again, the quiz is reset; otherwise, the application is closed.

8. Code Organization:

- The code is organized into methods and events for clarity and maintainability.
- GUI elements are manipulated based on the current state of the quiz.

In summary, this code creates a quiz application that guides users through a series of questions, provides instant feedback, and concludes with a final score. The application's GUI is designed to be user-friendly and interactive.